applied sciences

Article

Data Science Application for Failure Data Management and Failure Prediction in the Oil and Gas Industry: A Case Study

Simone Arena [1], Giuseppe Manca [2], Stefano Murru [2], Pier Francesco Orrù [1], Roberta Perna [1,2] and Diego Reforgiato Recupero [3,*]

[1] Department of Mechanical, Chemical and Industrial Engineering, University of Cagliari, Via Marengo 2, 09123 Cagliari, Italy
[2] Italteleco S.r.l., Via Edward Jenner, 19, 09121 Cagliari, Italy
[3] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
* Correspondence: diego.reforgiato@unica.it

Abstract: In the industrial domain, maintenance is essential to guarantee the correct operations, availability, and efficiency of machinery and systems. With the advent of Industry 4.0, solutions based on machine learning can be used for the prediction of future failures by exploiting historical failure data. Most of the time, these historical data have been collected by companies without a specific structure, schema, or even best practices, resulting in a potential loss of knowledge. In this paper, we analyze the historical data on maintenance alerts of the components of a revamping topping plant (referred to as RT2) belonging to the SARAS group. This analysis is done in collaboration with the ITALTELECO company, a partner of SARAS, that provided the necessary data. The pre-processing methodology to clean and fill these data and extract features useful for a prediction task will be shown. More in detail, we show the process to fill missing fields of these data to provide (i) a category for each fault by using simple natural language processing techniques and performing a clustering, and (ii) a data structure that can enable machine learning models and statistical approaches to perform reliable failure predictions. The data domain in which this methodology is applied is oil and gas, but it may be generalized and reformulated in various industrial and/or academic fields. The ultimate goal of our work is to obtain a procedure that is simple and can be applied to provide strategic support for the definition of an adequate maintenance plan.

Keywords: predictive maintenance; data analysis; failure prediction; machine learning; data cleaning; time series

check for updates

Citation: Arena, S.; Manca, G.; Murru, S.; Orrù, P.F.; Perna, R.; Reforgiato Recupero, D. Data Science Application for Failure Data Management and Failure Prediction in the Oil and Gas Industry: A Case Study. *Appl. Sci.* **2022**, *12*, 10617. https://doi.org/10.3390/app122010617

Academic Editor: Giacomo Fiumara

Received: 5 October 2022
Accepted: 14 October 2022
Published: 20 October 2022

1. Introduction

In the fourth industrial revolution, companies are constantly looking for ways to increase production and reduce costs, especially those deriving from maintenance, which amounts to about one-third of total operating costs [1].

In this scenario, data science is having great success because it can be leveraged to extract valuable information from data that companies can exploit to their advantage to expand their business and evolve [2]. However, not all companies are still able to carry out this process, and knowing how to maximize the employment of self-learning models represents one of the main issues linked to the process of extracting information from data [3].

Data science [4,5] is giving remarkable results in various scientific fields, such as materials science [3], maintenance management [6], programming [7] and improving the quality of a product or service performed [8]. Thanks to the continuous development of hardware, cloud systems, and new technologies, there has been an incentive for the increasingly frequent use of machine learning techniques [9]. This phenomenon is also supported by:

- The growing availability of open source software packages that make the implementation of self-learning models simple [10,11];
- The improvement of the availability of industrial data, big data, and servers in the data processing phase [12,13].

Furthermore, within the manufacturing systems, new, reliable and secure human-centered architectures have recently appeared at the heart of the evolution towards Industry 5.0 [14].

The employment of data processing techniques to carry out predictive analysis of failures in the industrial sector has become a particularly hot topic due to the need to obtain a maintenance strategy that guarantees the availability of the systems and prevents unscheduled downtime due to sudden failures [2,3]. Furthermore, in very competitive environments, it is necessary to better manage delivery times and reduce costs associated with non-compliant products that must be reprocessed, if possible, or thrown away [15].

During the life cycle of an industrial machine, its components could be damaged and broken, leading to high costs that companies have to face. For this reason, it is important to diagnose the roots of failure over time. Traditionally, these diagnoses have been made based on criteria (manual or knowledge-based) established by experts, but sometimes they may not work. It follows the need to study the data collected by the machines using sensors and identify possible operating anomalies to predict an imminent failure and intervene to avoid the consequences [16].

There are several types of historical components in petrochemical plants. For example, we could have rotating machines such as pumps and compressors. For these, in the literature, we have examples of failure prediction approaches that exploit data extracted from different sensors to provide adequate information for failure prediction [1,9,10,16–19].

In the literature, there are three maintenance approaches [1,20]:

- Run to failure (R2F). This strategy is based on post-failure interventions and is therefore the simplest of the maintenance approaches. Its use is justified when the consequences of the failure lead to the replacement of the single component without causing damage to the adjacent equipment and, more importantly, the failure must not have an impact on the environment and human safety.
- Preventive maintenance (PvM). In this strategy, we intervene through a plan of periodic corrective actions. This largely avoids breakdowns, but unnecessary maintenance is often carried out, leading to an inefficient use of resources and increased operating costs. PvM is used in cases when a failure can cause significant damage to the safety of workers and the environment. Furthermore, the consequences of the breakdown can lead to a long period of prolonged disruption and/or even serious damage to adjacent components.
- Predictive maintenance (PdM). Maintenance is performed based on the estimated health of the component. The PdM systems allow for detecting the fault before its occurrence, thus giving the possibility to intervene in advance. All this is possible thanks to historical data, statistical methods, machine learning and engineering approaches. In particular, machine learning techniques help defining useful methods on the data for predicting future failures and for intervening proactively. This maintenance strategy tries to overcome the limitations of PvM by trying to exploit the component until the end of its life. It is therefore used for particularly critical components due to their high cost of use and for possible losses in terms of safety.

In this work, we will focus and employ a predictive approach based on data analysis and machine learning techniques to predict future failures. The case study is represented by the analysis of the time series of the failures of the components of a revamping topping plant (RT2) belonging to the SARAS group. It will show which issues present the raw data and which pre-processing techniques we have used to clean and complete the data with the goal of having a number of features and different samples that can be organized in a data structure to be fed to the predictive models. Indeed, in real-world applications, the dataset that it has to be managed can get extremely massive, complicated, and incomplete, and mostly, it can contain data of all sorts, including missing or inconsistent values. Therefore, data pre-processing is a crucial step in the implementation of machine learning algorithms to extract valuable information, making the database completer and more precise. Thus, the achieved quality of the well-structured dataset makes the ML algorithm able to accurately predict future observations. Simple natural language processing (NLP) approaches are used to categorize each occurred fault and thus create a taxonomy that can be used to classify future failures. The goal is to forecast the next failure for each component in the RT2. In particular, predictions will be made by exploiting the time series of the past related to single-component failures using simple algorithms such as linear regression, ARIMA, and a polynomial interpolation strategy we propose. ARIMA has already been widely used in the predictive field to make predictions on time series [21] and has already been used in the petroleum environment to make predictions on the performance of crude oil wells [22].

The remainder of this paper is organized as follows. Section 2 shows examples of works similar to ours and the differences between our proposal and the state of the art. In Section 3, the industrial plant that we have considered for its failure data will be described. Section 4 contains the description of the historical data that will be processed. In Section 5, we will explain the methodology with which data have been analyzed and fed to different machine learning techniques. In Section 6, the results obtained using the methodology developed in Section 5 will be shown. In particular, the results will be illustrated, step by step, starting from the reconstruction of the data and ending with the forecasts of the faults. Finally, Section 7 ends the paper with conclusions drawn as a function of the obtained results, discussions on best practices that we define, and future directions we are headed toward to improve the performances we obtained and generalize our approach.

2. Related Work

Currently, data are continuously generated as a result of the digitalization of manufacturing systems, enabling thorough data analytics. Large databases frequently have a significant percentage of unreliable and irrelevant information due to noise, missing data, data redundancy, or insufficient labelled data. Particularly, data-driven methods often deal with difficulties in industrial applications where real data are collected from multiple heterogeneous sources. In this scenario, data preprocessing is a crucial stage in data analysis and has a significant impact on modelling outcomes. It represents a must-do process before data analysis is performed, mainly involving four processes: cleaning, integration, transformation, and reduction of raw data [23] aiming at improving the dataset quality and the prediction accuracy.

Over the years, several methods and procedures for handling incomplete datasets have been proposed in the literature. A manual search selection to screen suitable works within the scope of the proposed paper is carried out by focusing on reviews concerning preprocessing techniques applied to machine learning approaches in the manufacturing industry.

Thus, Ning et al. [24] compared three different algorithms for oil production forecasting: auto-regressive integrated moving averages (ARIMA), long short-term memory (LSTM) network, and Prophet. Singh et al. [25] presented a comprehensive review for fault detection and diagnostics in heating, ventilation, and air conditioning (HVAC) systems, while Dogan and Birant [26] and Zhu et al. [23] focused their review on manufacturing applications. Seo et al. [27] proposed a methodology for managing missing values of dissolved gas analysis (DGA) data. Wang et al. [28] proposed a combined system based on data preprocessing and multi-objective optimization in a wind production application. Ranjan et al. [29] presented a review of preprocessing methods for data from power system applications, while He et al., Xiao et al., and Xiao et al. [30–32] developed a data preprocessing approach for energy consumption. Frye et al. [33] reported a benchmark-based approach to be applied to production systems.

By analysing the extant literature on data preprocessing, it can be noticed that it is recognized as a crucial step since well-structured data can provide significant knowledge, considered a valuable resource in the decision-making process [34].

It is worth highlighting that the challenge in ensuring a large amount of high-quality data is related to settling the proper assignment of processes, data-oriented decision making, more efficient operations, and reduced risks [35]. In this case, a data-driven approach is applied to prevent the breakdown of equipment in advance via unique identifiers. This allows enhancing the accuracy of diagnosis aiming at developing suitable failure predictive models. In this scenario, the adoption of machine learning (ML) techniques is capable of performing prognostics and failure prediction tasks [36].

Currently, PdM-related solutions based on ML techniques are widely investigated for industrial applications such as manufacturing, energy production, transportation, and heavy and light industry, among others. Therefore, the studies based on the real-world dataset are limited, as reported by Leukel et al. [11] and Carvalho et al. [37], who presented a recent systematic review on the use of ML technology for failure prediction in industrial maintenance. However, restricting the analysis to operating machinery applications, the sample size is further reduced [1,12].

Liu et al. [17] presented a comprehensive review of fault diagnosis approaches based on artificial intelligence algorithms for rotating machinery. Hajizadeh [38] and Hanga and Kovalchuk [39] proposed fault detection techniques based on artificial intelligence in the oil and gas industry. Qian et al. [40] and Zhang et al. [41] investigated the transfer learning method for rotating machinery by considering variable working conditions, while Yang et al. [42] proposed a scheme based on hierarchical symbolic analysis (HSA) and a convolutional neural network (CNN).

In [18,19,43,44], the authors proposed different ML approaches to be applied to roller-bearing fault diagnosis. Orrù et al. [1] compared two ML algorithms based on support-vector machines (SVM) and multi-layered perceptron (MLP) by using a real dataset from sensors placed on a centrifugal pump. Erdem and Eken [45] and Breviglieri et al. [46] applied a deep learning model for a smart-grid stability prediction. In [47–49], the authors used an SVM algorithm for failure detection, while Jirdehi and Rezaei [14] proposed two different techniques based on artificial neural networks (ANN) and adaptive neuro-fuzzy inference systems (ANFIS).

An ANN model was also presented by Guedes et al. [50] to monitor the actual condition of the stator electrical insulation of three-phase induction motors and by Yu et al. [51] to predict failures of an oil-immersed transformer. Paolanti et al. [52] presented a random forest model for fault diagnosis of a cutting machine. Giantomassi et al. [53] adopted an estimation algorithm of the probability density function of current signals for the failure detection and diagnosis of an electric motor. Chen et al. [54] developed an approach based on the combination between CNN and extreme learning machine (ELM) to be applied to a gearbox and motor bearing. In [55–57], the authors adopted ensemble learning models for different rotating machinery. Khorsheed and Beyca [58] proposed a framework for the proper implementation of an ML-based PdM strategy for pumping systems. Tang et al. [59] reported a novel adaptive learning rate deep belief network combined with Nesterov momentum applied to a gearbox and locomotive bearing test rigs. Zanisek et al. [60] and Lee et al. [61] proposed a machine learning-based approach for the health condition monitoring of a radial fan impeller and machine tool system elements, respectively.

The reported literature review depicts a scenario where machine learning turns out to be a very appealing field of research, especially for industrial maintenance management since it emerges as a powerful resource for the development of intelligent predictive algorithms [62]. However, ML adoption in real factories is still considered challenging mostly due to several barriers that limited its implementation regarding current industrial applications [11,26,63]. Some of the main challenges are: (i) the acquisition of a suitable and relevant dataset since it strongly affects the predictive model efficiency and (ii) the selection of a proper ML algorithm. The former challenge highlights the importance of preprocessing data as a vital process impacting the results, especially when real raw data are used, while the latter assumes that each manufacturing problem is unique, and each algorithm performs differently depending on the data available, as well as parameter settings. Therefore, this requires the analysis of the different models, aiming at selecting the one capable of solving the targeted manufacturing problem. To deal with these open challenges, this work proposes a preprocessing technique for the historical data of failures of different plant equipment operating in the oil and gas sector. Moreover, the achieved structured dataset is used to develop machine learning models and statistical approaches aiming at performing reliable failure predictions to support the maintenance decision-making process.

3. Description of the Plant

The revamping topping plan (RT2) is one of the crude oil fractionation plants owned by the SARAS group. This plant is fed with crude oil transported by ship. The crude oil is classified into the following products:

- Propane;
- Butane;
- IC5 + light petrol;
- Heavy gasoline;
- Kerosene;
- Light diesel;
- Heavy diesel.

The RT2 plant consists of two main sections:

- The atmospheric distillation section;
- The petrol splitting and liquefied petroleum gas (LPG) recovery section.

The atmospheric distillation section has the purpose of dividing the crude oil into the various products listed above. It is also equipped with compressors for gas. This section includes the following subsections:

- Heat recovery (exchange train). The preheating of the crude oil is performed in the exchange train, consisting of a group of heat exchangers, at the expense of the heat of the products and of the circulatory reflux of the distillation plant.
- Desalination. The crude oil, after being preheated, is mixed with a small percentage of water and is sent to the desalter. Here, due to an electric field, the hydrocarbon phase is separated from the water phase, rich in salts, which is discharged. This greatly reduces the residual salt content in the crude oil.
- Furnaces. The final preheating of the crude is obtained through two vertical cylindrical furnaces, each consisting of a convective section and a radiant section. Before entering the convective area, the blank line is divided into several coils with flow regulation to ensure a balanced distribution of the flow. After recovering the heat of the combustion fumes, the raw material undergoes a final temperature increase due to the radiation.

The raw material coming from the furnaces, partially vaporized, is transferred to the distillation column, where it is divided into different cuts. In particular, the following products are obtained:

- Residue: after being stripped in a counter-current with steam, the residue is extracted, under level control, from the bottom of the column, and transfers heat to the raw material in the exchange train and in some reboilers of the light fractionation section. Then, it is refrigerated in tempered water exchangers and sent to storage.
- Heavy gas oil: stripped in a special column, the heavy gas oil transfers heat to the crude oil in the exchange train, is cooled in tempered water coolants and is sent to storage under flow control.
- Light diesel: stripped with steam, the light diesel exchanges heat with crude oil, is cooled in refrigerants and sent to storage in flow control and/or to the mild hydrocracking plant (another catalytic type treatment plant) and/or to the diesel drying column.

- Kerosene: stripped with steam, the kerosene transfers heat to the raw material in the exchange train. It is cooled in air refrigerants and sent to storage in flow control.
- Top product: This fraction includes the light constituents of the crude oil, gasoline, condensable gases, and non-condensable gases (fuel gas). It is cooled in two successive stages at different temperature values: the liquid product is sent back to the column, while the condensate is sent to the fractionation section, and the gases are conveyed to a compression station for the recovery of the LPG fraction outside the battery limit of the system.

In the second part of the plant, named the petrol splitting and liquefied petroleum gas (LPG) recovery section, a second fractionation takes place to obtain light fuels. First of all, in the stabilizer column, the separation of gases (top product) from petrol (bottom product) takes place. The gases, LPG and non-condensable, refrigerated and condensed, are partly recirculated and partly sent to another plant for the recovery of the LPG fraction. The petrol, after heat exchange, undergoes a further fractionation in a special splitting column. As for the stabilizer, the heat at the bottom of the column is supplied by two reboilers, which exchange with the bottom residue of the main column. Subsequently, there is the petrol splitter, which receives the stabilizer bottom product in charge after preheating and divides it into three cuts:

- Overhead product: this consists of a mixture of light petrol and isopentane. After condensation, it is partly recirculated in the column and partly sent to storage. From there, it is sent to other plants for separation into the two components.
- Lateral cut: this is a medium-density petrol which, after heat exchange with the charge to the column, is sent to storage.
- Bottom product: this is composed of the heaviest fraction of petrol and sent to storage after having transferred heat to the charge and having been further refrigerated. The heat necessary for fractionation is supplied to the bottom product through two reboilers, which exchange with the upper circulatory reflux.

Further details about the plant can be found in private corporate documents [64–66].

4. Data Description

The data we have worked with are related to the RT2 plant belonging to the SARAS group and consist of "Notices" and "ODM" (maintenance orders):

- The first one contains the chronology of all the anomaly notices that occurred for each component of the RT2 system from 2000 to 2021; we will refer to these also as alerts or notices;
- The second contains the maintenance orders for the same components, that is, the order of the materials needed for maintenance activities.

We use "ODM" to understand whether a certain notice corresponds to a breakdown or a simple malfunction (a breakdown is related to a piece of equipment that has broken, whereas a malfunction corresponds to a piece of equipment or machinery that fails to function normally but has not broken yet).

Both "Notices" and "ODM" have a similar structure: each sample (row) corresponds to a notice with different features (columns) about the notice. The features of interest for our analysis are:

- The identification number of the notice;
- The description;
- The date of the event;
- The technical office;
- The equipment.

The identification number is a unique numeric code necessary to understand which notice is being referred to. It is used to link information about a certain notice from "Notices" to "ODM". The description includes a short sentence indicating what happened. The date of the event represents the moment of occurrence of the event, which may be a breakdown or malfunction. The technical office defines the place where a specific anomaly occurred. The purpose of the technical office is to indicate the component that has presented a malfunction or breakdown and is represented by abbreviations that can be composed of three or four elements in the form:

```
"Plant" - "class of the component" - "component-name of the component"
```

or:

```
"Plant" - "class of the component" - "name of the component" -
"part of the component"
```

An example for the first case is represented by:

```
RT2-FURNACES-RT2F1A
```

where FURNACES corresponds to "furnaces" and RT2F1A is a component of the class "furnaces". An example for the second case is:

```
RT2-PUMPS-RT2MP4-RT2MP4P
```

where PUMPS represents the class of pumps, RT2MP4 is the associated component and, finally, RT2MP4P is a sub-part of the component RT2MP4.

The last feature of the "Notices" and "ODM" data, the equipment, if indicated, represents the part of the component as shown in the second form of the technical office just mentioned.

The main problem with the "Notices" data lies in the "technical office" column, which is usually blank or partially filled. This generates an enormous amount of incomplete information that cannot be used in any way. As the technical offices are supposed to provide the information about the offices and the components that have presented a possible anomaly, without that, we cannot associate the corresponding failure data. Therefore, many components do not have information about their corresponding technical office, making them unsuitable for the use of machine learning techniques.

This problem occurs because the compilation of the alerts has been performed manually under the responsibility of the operators. Sometimes the operators omitted text in this field, compromising the overall quality of the available data.

Thus, one problem we had to face was the reconstruction of the missing data. This was performed with a pre-processing step with the goal of filling the missing technical office names and reorganizing the overall structure so that each component may be defined and associated with a chronological data set of its notices.

To reconstruct the data, first we had to identify the columns from the "Notices" data that could be a source of information or that could indicate which component presented an anomaly. In particular, the column "Description" present only in "Notices" and "Equipment" present both in the "Notices" and in the "ODM" have been considered. To reconstruct the missing data, the "Description" and the "Equipment" fields could give some clues. During the reconstruction, an attempt will be made to reconstruct the technical offices at least up to the third element, which indicates the component that has presented a malfunction or breakdown. The way we have exploited the "Description" and the "Equipment" features to reconstruct the technical offices up to the third element and how we carried out the entire process is detailed in Section 6.1.

The "Notices" data contain 10,700 notices, 10,567 of which have an incomplete or missing "Technical office". According to the "ODM" data, it turned out that 4,907 notices out of 10,700 correspond to a fault, while the remaining 5,793 correspond to malfunctions. In total, 30 classes of components will be analyzed, for a total of 1,036 components. In Table 1, we show three classes of examples and their corresponding components. The reader notices that the class in the first row, FURNACES, corresponds to "furnaces" and includes two components: RT2F1A and RT2F1B. The element in the second row, COMPR, corresponds to "compressor" and includes two different components: RT2MC1 and RT2MC1B.

Finally, the element in the third column, COLON, represents the “distillation columns” and includes six further elements: L1T3, L1T4, RT2T1, RT2T2A, RT2T2B, RT2T2.

Table 1. Example of three component classes and the elements they include.

Class	Components
FURNACES	RT2F1A, RT2F1B
COMPR	RT2MC1A, RT2MC1B
COLON	L1T3, L1T4, RT2T1, RT2T2A, RT2T2B, RT2T2C

5. Proposed Approach

The architecture of the approach we propose is depicted in Figure 1 and consists of four main modules:

- Data reconstruction module;
- Module for categorization of maintenance alerts;
- Data preparation module;
- Prediction module.

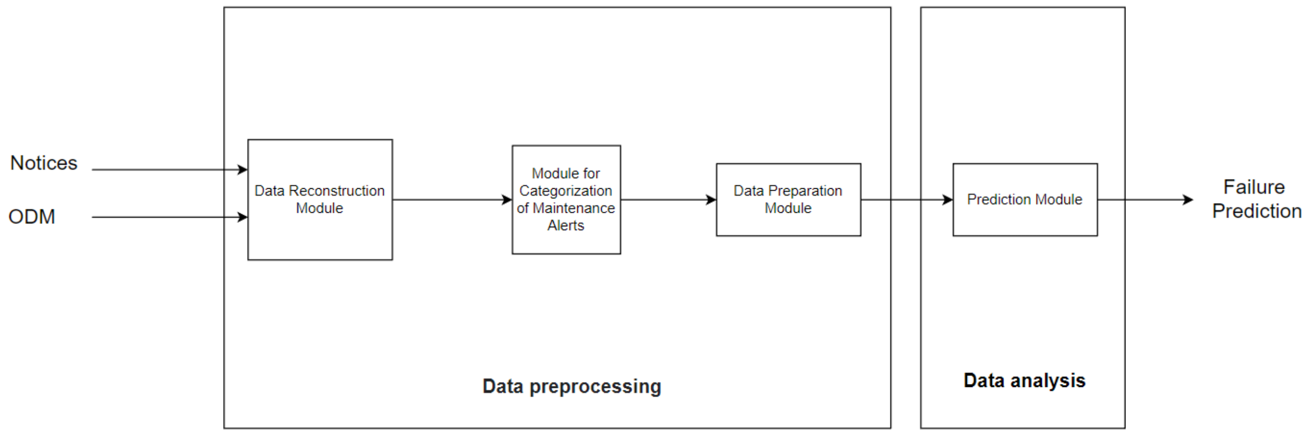

Figure 1. Flow chart of the four main modules.

As already mentioned, the provided data are characterized by a huge amount of unstructured data lacking the most important information, namely the name of the component that has failed.

For the first module, it is necessary to define support data that is useful for the reconstruction step. For such a purpose, we have used the genealogy of the plant components, where the plant name will represent the root, the type of the components will be internal nodes named NodetypeA, and the component names will be internal nodes named NodetypeB. Sometimes, a NodetypeB can have children that are considered leave elements. This last element specifies the part of the component. To build the link between the type of component and the respective component of the plant, we start with the names of the components, and we will create special “dictionaries”. A dictionary is a structure that contains key-value pairs. Three dictionaries will be defined to properly handle the structure of the components, as discussed in Section 5.1. The link between the key and the value will allow us to walk through the family tree to reconstruct the unstructured data and identify the location of the fault.

The first module will be described in Section 5.1, in which the data reconstruction techniques that we have employed will be shown. In particular, we are interested in reconstructing the “technical site”, which has the task of indicating which component has failed. For us, these data are important, because for each component, we are interested in extracting the time series of the related faults and using it in the forecast step. After reconstructing the data, we will proceed with the categorization of the alerts. This module will be described in Section 5.2. The categorization of the maintenance notices can be done by attributing to each notice a word deemed important, which may represent a fault phenomenon or a maintenance activity. This process may be important for future work on the prediction of faults. We may be interested not only in predicting the date of the failure but also the category of the component. This may provide useful information, such as the cause of the failure or suggesting what activities should be carried out following a failure and how much it could cost. As the last step, the total data will be reorganized by dividing the information of each component and sorting them in chronological order. In Section 5.3, the module dealing with the final data structure obtained through the previous steps and the reordering of the alerts will be shown.

Finally, in Section 5.4, the data pre-processing module to generate the input for the predictions will be shown, and the self-learning models used will be described. In this phase, the forecasts of the next failure will be calculated for all system components that have a sufficient number of historical data.

5.1. Data Reconstruction Module

First, it is necessary to define the family tree of the components. Each component belongs to a type that is part of the plant in question; therefore, we could think of building a tree where the main root represents the plant, the branches represent the various types of components and the leaves represent the components or the specific parts of them. Figure 2 shows a subset of the tree in which the genealogies of 3 types of components are reported. RT2 is the name of the plant, and COMPR, PUMPS and FURNACES are three classes referring to compressors, pumps, and furnaces, respectively. Then, we have components corresponding to NodetypeB, and, for some of them, we have specific parts of them at the leaf level.

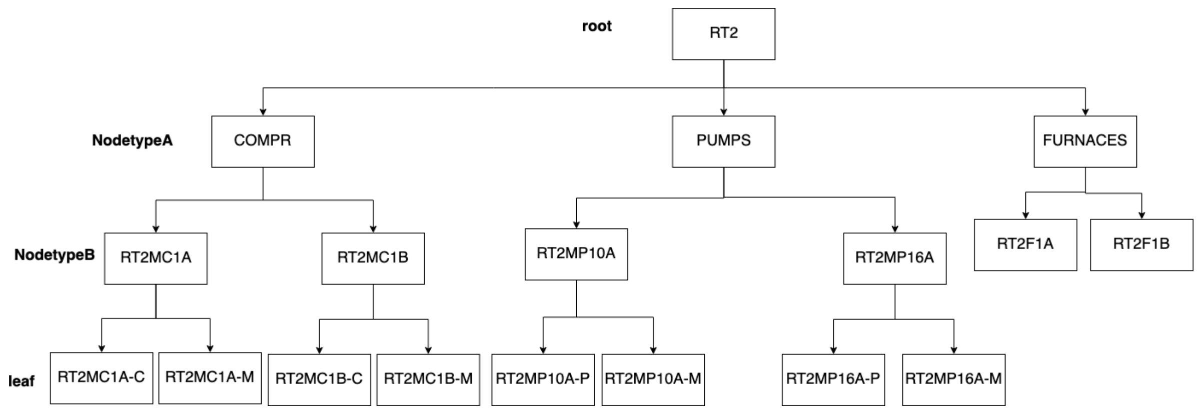

Figure 2. Subset of the component hierarchy.

We could, therefore, think of constructing the link between leaf and root, internal nodes and root, and direct link between leaf and root using a "dictionary". A dictionary contains a sequence of key/value pairs. Keys and values can be given of any type; in our case, we will use strings and lists.

A dictionary allows us to recreate the link between one part of the tree and the other, managing to recreate an abbreviation such as:

$$Root - NodetypeA - NodetypeB - leaf$$

This abbreviation is similar to that of the technical site where "ROOT" is the name of the plant, the "NodetypeA" element will be the type of component, the "NodetypeB" element will be the name of the component, and the "Leaf" will be the part related to the component.

We have defined three main dictionaries with the following key/values pairs:

- NodetypeA-NodetypeB: a dictionary that presents the types of components as keys and the list of the respective components as values for each key;
- NodetypeB-Leaf: in analogy to the previous dictionary, the system components will be the keys, and for each of them, we associate the list of their related mechanical parts;
- NodetypeA-Leaf: a dictionary that presents as keys the various types of components associated with the list of the mechanical parts of the related components.

During the reconstruction phase, we also made use of the corresponding reversed dictionaries, where keys and values were swapped. The use of these dictionaries allows, starting from a match between strings, to reconstruct the link between the elements in the technical area. To complete the missing information in the column of technical locations, five different methodologies have been defined:

1. Completion using the information contained in the "Equipment" column of "Notices";
2. Completion using the information contained in the "Equipment" column of "ODM";
3. Completion through the technical sites of "ODM";
4. Completion by mapping the words contained in the "Description" column and the names of components;
5. Search for components using the suffixes of the components contained in the "Description" field.

5.1.1. First Methodology

The first technique involves searching in each notice for the presence of a component in the "Equipment" column of "Notices". If found, the component is added to the technical site of the respective row to correctly reconstruct the technical site. The following alternatives may arise:

A. In the technical area, there is only ROOT;
B. The technical office is constituted by ROOT-NodetypeA;
C. In the technical site, there is ROOT-NodetypeA-NodetypeB.

In case A, if a leaf element is present in the "Equipment" column, it is searched in the dictionaries. The "NodetypeA" and "NodetypeB" elements are extracted, and the technical site is rebuilt by placing the various elements in the correct order. There is no need to search for the "ROOT" element because the system is unique and is common to all technical sites.

In case B, if a leaf element is present in the "Equipment" column, it is searched in the dictionaries, and only the "NodetypeB" element will need to be extracted. As in the previous case, the elements "NodetypeB" and "Leaf" are added in the technical site to complete the missing information.

In the end, for case C, if in the "Equipment" column, there is a "leaf" element, it is added to rebuild the technical site.

Table 2 shows the examples corresponding to the described cases. In the top row (case A), a leaf element is present in the equipment field. By checking the dictionaries, we can retrieve the entire chain up to the root RT2, RT2-PUMP-RT2MP23A-RT2MP23A-P. In the middle row (case B), the leaf element is present in the equipment, and, according to the dictionaries, if it is a child of the type COMPR, then it is added to the reconstructed technical site together with its parent. Finally, the last row (case C) has a value within the equipment. This is searched within the dictionaries, and it is found as a leaf matching the information reported on the technical site. It is therefore added to the rebuilt technical site.

5.1.2. Second Methodology

The second heuristic is very similar to the first one, with the difference that it looks for the leaf component within the "Equipment" column of the "ODM" data and not the "notices" data. Examples for such a heuristic are the same as those shown in Table 2, with the only difference that the equipment column is chosen from the "ODM" data instead of the "Notices".

Table 2. Examples of reconstruction data with the first methodology.

Number of Notice	Description	Old Technical Site	Equipment	Technical Site Rebuilt
10,026,296	Cleaning filter and suction line mp23a	RT2	RT2MP23A-P	RT2-PUMSP-RT2MP23A-RT2MP23A-P
10,027,059	SYSTEM CONTROL MC1A	RT2-COMPR	RT2MC1A-C	RT2-COMPR-RT2MC1A-RT2MC1A-C
10,013,510	Large leak	RT2-PUMPS-RT2MP13A-	RT2MP13A-P	RT2-PUMPS-RT2MP13A-RT2MP13A-P

5.1.3. Third Methodology

The third method consists of comparing two technical sites coming from two different data ("Notices" and "ODM") with the same identification number of the notice. If the technical office in ODM contains additional information compared to the one in "Notices", these data are copied to reconstruct the new technical site. An example of this methodology is shown in Table 3.

Table 3. Example of reconstruction of the technical site using the third methodology.

Technical Site in "Notices"	Technical Site in "ODM"	Technical Site Rebuild
RT2-PUMPS-RT2MP4	RT2-PUMPS-RT2MP4-RT2MP4_P	RT2-PUMPS-RT2MP4-RT2MP4_P

5.1.4. Fourth Methodology

The fourth heuristic involves analyzing each sentence contained in the "Description" column using the following actions:

- The sentence is split into a list of tokens;
- If in the technical site there is a NodetypeA or NodetypeB, the list of components or parts of components having that as father is extracted;
- Each token extracted from the description is compared with the names of the components extracted in the previous step;
- If a token in the description coincides with a component name mapped among those extracted, then it is added to the technical area.

The code is structured in different levels of reconstruction depending on the number of information contained in the technical site:

1. If the technical site includes just RT2, the description value is examined to find any element descendant of RT2 (child or descendant). Three things can happen:
    A. If the element found in the description is NodetypeA, it is therefore added to the technical site, thus obtaining:

    $$ROOT - NodetypeA$$

    B. If the element found in the description is NodetypeB, the dictionary structure is queried to find the NodetypeA between the ROOT and the found NodetypeB. Subsequently, NodetypeA and NodetypeB are added to the technical site, obtaining a value of the type:

    $$ROOT - NodetypeA - NodetypeB$$

    C. If the element found in the description is a leaf, the dictionary structure is accessed to find the NodetypeB father of the leaf. Once we find it, we look for the NodetypeA that links it to the ROOT as we did in the previous case. Then, the elements NodetypeA, NodetypeB and the leaf are added to the technical site, thus obtaining:

$$ROOT - NodetypeA - NodetypeB - Leaf$$

2. If ROOT-NodetypeA is present in the technical site, we first extract the values from the dictionary having the NodetypeA element as a key. Subsequently, each token of the description is compared against the elements extracted from the dictionary, and, if a match is found, the technical site is completed with the matched element. It can either be just a NodetypeB or both a NodetypeB and a leaf element. The technical site will therefore have the form:

$$ROOT - NodetypeA - NodetypeBorROOT - NodetypeA - NodetypeB - Leaf$$

3. If ROOT-NodetypeA-NodetypeB is present in the technical site, in this case, firstly the values from the dictionary having NodetypeB as the key are extracted. Then, each token of the description is compared against the extracted elements, and in case a match is found, the technical site is completed with the leaf element and will therefore have the following form:

$$ROOT - NodetypeA - NodetypeB - Leaf$$

In Tables 4–6 the examples related to the three different aforementioned cases of the fourth methodology are depicted. In particular, in the description of the first row of Table 4, the term pumps is found. It corresponds to the pump type, which is a direct child of the root. The new technical site becomes RT2-PUMPS. For the second row, the token level corresponds to a NodetypeB whose ancestors are VALV and RT2. For the third row, the word r2frcv62 in the description corresponds to the leaf element with PORTATA as NodetypeA and VALV as NodetypeB. Table 5 includes in the first row the RT2MC1A element, which is a NodetypeB with the same class component (COMPR) as that found in the current technical site. The table has in the same row the element MP19A-P, which is a leaf with corresponding NodetypeB RT2MP19A. This has PUMPS as NodetypeA, the same as in the current technical site, which is then augmented as reported in the table. Similar mechanisms are applied in Table 6.

Table 4. Example of reconstruction site using the fourth methodology when the technical site contains just the ROOT element.

Number of Notice	Description	Old Technical Site	New Technical Site
10,011,330	RT2 pump—CHECK the tooth joint	RT2	RT2-PUMPS
10,013,025	RT2 level CONTROL lrc1	RT2	RT2-VALV-LEVEL
10,010,917	replace intercept r2frcv62	RT2	RT2-VALV-FLOW-R2FRCV62

Table 5. Example of reconstruction site using the fourth methodology when the technical site contains ROOT-NodetypeA.

Number of Notice	Description	Old Technical Site	New Technical Site
10,014,245	RT2MC1A cleaning CRANK OIL FILTER	RT2-COMPR	RT2-COMPR-RT2MC1A
10,029,113	MTZ RT2 MP19A-P	RT2-PUMPS	RT2-PUMPS-RT2MP19A-RT2MP19A-P

Table 6. Example of reconstruction site using the fourth methodology when the technical site contains ROOT-NodetypeA-NodetypeB.

Number of Notice	**Description**	**Old Technical Site**	**New Technical Site**
10,013,299	CHECK functionality L1FRC56	RT2-SPORT-FTDPE	RT2-SPORT-FTDPE-L1FRCT56

5.1.5. Fifth Methodology

The fifth methodology of completion is applied only to notices that in the technical site refer to a NodetypeA element (defined as an instrument) of the following type:

$$Instruments = [SLIV, SPORT, SPRES, STEMP, SVARI, VALV, VALVR]$$

The rationale behind this is that instruments (such as pressure gauges, valves, etc), unlike the other system components, are mounted on the same loop and usually have the same final numerical value, which represents the number of the loop they are part of. This methodology leverages the numerical value they have as a suffix in their name, trying to reconstruct further technical sites because of that.

In this case, we proceed as follows:

1. We check if we find a NodetypeA element in the technical site that corresponds to an element of the list defined above;
2. Then we consider the last element of the technical site (which might be different than the NodetypeA element found in the previous step) and look in the dictionary for leaves having as their ancestor the last element of the technical site, which can be either a NodetypeA or NodetypeB element.
3. We perform a search in the current description for tokens having at least one internal numerical value (surrounded by other symbols); a token satisfying such a constraint is considered a component by ITALTELECO.
4. If such a component is found, we take its last number from its string name;
5. This number is used to search for a component from the list found at step 2 that has as a suffix the identified number;
6. In the case of correspondence, there are two possibilities:
   A. ROOT-NodetypeA is already present on the technical site, and the component found is a leaf. Consequently, the father (NodetypeB) of the leaf is searched through the dictionary structure and will be included in the technical site, obtaining a structure such as:
   $$ROOT - NodetypeA - NodetypeB - Leaf$$
   B. ROOT-NodetypeA-NodetypeB is present on the technical site. In such a case, the component found is a leaf. Therefore, it is only necessary to add the element to the technical site, obtaining:
   $$ROOT - NodetypeA - NodetypeB - Leaf$$

Some examples related to the fifth methodology are illustrated in Table 7. The first row has VALVR as NodetypeA and therefore satisfies the first step. Then, we extract the leaf elements descendants of VALVR (being the final element of the technical site). Among others, RT2PV7 is one such element. In the description, we find RT2PCV7C as a token satisfying step 3 and identify the number 7 from step 4. Among the leaf elements found in step 2, RT2PV7 satisfies the constraints in steps 5 and 6. The example shown in the second row of Table 7 works similarly.

Table 7. Example of reconstruction data applying the fifth methodology.

Number of Notice	Description	Old Technical Site	New Technical Site
10,336,724	RT2PCV7C DISASSEMBLY REGULATORS	RT2-VALVR	RT2-VALVR-PRESS-RT2PV7
10,162,193	RT2FRC22 function check (T1 column)	RT2-SPORT-FE	RT2-SPORT-FE-R2FE22

5.2. Module for the Categorization of Maintenance Alerts

In this module, we want to classify the notices according to certain classes and using words that we need to define. These classes, which we will call "Fault Categories", can represent a failure phenomenon or a maintenance activity. Classifying the notice according to these "Fault Categories" can help understand which failure phenomena or maintenance activities are more frequent for each component type. Additionally, this information may be used as a predictive variable. First, the words contained in the "Description" were analyzed to identify which ones were the most recurrent overall and grouping components in categories. We used a list of "stop words" consisting of words that are not likely to provide information about the faults such as articles, prepositions, and conjunctions. When breaking down the "Description" in tokens, if a token was a stop word, we simply removed it. Besides common stop words like those mentioned above, other stop words have been defined in collaboration with ITALTELECO. When we considered all the components independently, the following steps were followed:

1. Only the Descriptions that in the technical site have at least ROOT-NodetypeA are considered;
2. Each Description is broken down into tokens;
3. The obtained tokens are inserted into a list;
4. The output is represented by the frequency value of each word. The percentage of occurrence of each word was calculated using the formula:

$$presencepercentage = (numberofoccurrencesofthesingleword * 100)/(totalnumberofwords)$$

When considering the components grouped by typology, the following steps were carried out:

1. The "Notices" data were divided into several subgroups, each related to the type of components (which is basically the NodetypeA element present in the technical site);
2. For each subgroup of notices, points (a) to (e), defined in the previous case, were performed, obtaining the frequency value of each word (which this time is referred to each type of component).

The obtained results were shown to the ITALTELECO company, which took care of manually choosing the words of interest by identifying 19 "fault categories" which are depicted in Table 8. These words partly represent failure phenomena, such as vibrations or leaks, or maintenance actions, such as cleaning, adjustment, and replacement.

Each original word that we have provided has been mapped to one of the 19 categories. If a category includes more words, it means that they are all related or have the same meaning or are synonyms. They represent the final categories used to group the alerts. Clustering was performed to group together similar words or verbs in different forms. For example, words such as "cleaned", "clean", and "cleaning" are all mapped to the same category CLEANING. Once we had this list of 19 fault categories, we had to associate each alert with one of them. Therefore, this further categorization process that has been carried out dealt with checking each "Description" field to identify which fault class it contained.

Table 8. Fault categories.

	CATEGORIES
1	BLOCK
2	CONTROL
3	CORROSION
4	DAMAGE
5	REMOVE
6	FUNCTIONALITY
7	LUBRICATION
8	MANEUVERABILITY
9	MAINTENANCE
10	LEAK
11	CLEANING
12	REGULATION
13	REVISION
14	REPAIR
15	RAPTURE
16	NOISE
17	REPLACEMENT
18	WEAR AND TEAR
19	VIBRATION

At this point, we processed the “Notices” data by creating a dictionary having as keys the categories and as values the words present in the description corresponding to the categories (synonyms, verbal forms, plurals). This dictionary was created in collaboration with the company.

Once we defined the fault categories, the categorization process involved the following points:

1. Creating a new column in the “Notice” data called “Fault Category” initially empty;
2. For each notice, extracting the description;
3. Breaking the description down into tokens;
4. Comparing each token with the elements corresponding to each fault category extracted from the dictionary;
5. In case of a match with an element corresponding to a certain fault category, this was written in the underlying notice.

5.3. Data Preparation Module

The collection of maintenance notices needs to be rearranged to obtain a data structure that can be processed to extract valuable information. Since the notices for each component must be examined, the entire structure has been divided into subgroups, where each group is related to a component. Subsequently, each subgroup of notices was rearranged in chronological order of occurrence. The data structure for each component has the following items:

- The name of the component;
- The notice number;
- The date when the fault occurred;
- The fault category;
- The maintenance cost.

For example, in Table 9, a subgroup related to the component RT2MP5B-P, which represents a pump having eleven faults on eleven different dates, is reported. On the second-last line, the box relating to the fault category is empty. This may happen because the description related to that notice does not contain the information necessary to categorize the alert automatically.

Table 9. Example of standard structure obtained for the pump RT2MP5B-P.

Component	**Notice**	**Description**	**Date**	**Cost [€]**	**Fault Category**
RT2MP5B-P	10,046,792	PUMP COOLING H2O LINE OFF	27/12/2001	203.84	CLEANING
RT2MP5B-P	10,111,379	RT2MP5B CLEAR CIRCH20 COOL	09/03/2004	203.84	CLEANING
RT2MP5B-P	10,124,210	RT2MP5B mechanics control (LOSS)	30/08/2004	6882.88	CONTROLL
RT2MP5B-P	10,303,874	RT2MP5B RESTORE COOLING H2O	10/05/2010	549.8	MAINTENANCE
RT2MP5B-P	10,321,419	RT2MP5B RESTORE COOLING H2O	25/10/2010	200.32	MAINTENANCE
RT2MP5B-P	10,403,663	RT2MP5B SEAL REVIEW	10/12/2012	2605.54	REVISION
RT2MP5B-P	10,494,752	RT2 MP5B H2O COOL DESCENDING	06/07/2015	855.54	CLEANING
RT2MP5B-P	10,525,298	RT2 MP5B H2O COOLING Clogged	13/06/2016	597.68	CLEANING
RT2MP5B-P	10,528,736	RT2MP5 BLIND DISCS REMOVAL	20/09/2016	11,650.99	CLEANING
RT2MP5B-P	10,640,333	RT2 MP5B NO WATER CIRCULATION TXT	29/11/2019	2005.73	*none*
RT2MP5B-P	10,648,669	RT2MP5B H2O COOLING Clogged	03/03/2020	672.48	CLEANING

5.4. Prediction Module

The predictive models that will be presented below are all fed with the same input data and the prediction phase always follows a similar process. Therefore, in this section, we will describe how the data are fed to the predictive models with the goal of obtaining a forecast of the next fault for each component.

First, for each component cm we extract the fault dates of all the notices about c and put them in a vector in chronological order. Let us call such a vector "y". Subsequently, an "X" vector will be generated containing integer values from 1 up to the length of the vector "y". To work with integer values, the dates have been transformed into an integer representing the number of days elapsed from the oldest date in the "Notice" data; that is 31/12/1999. Some examples of conversion are shown in Table 10:

Table 10. Examples of conversion from dates to days with respect to 31/12/1999.

Date	**Converted Date (Elapsed Days)**
01/01/2000	1
14/09/2007	2814
15/05/2012	4519
06/07/2020	7493

The vectors "X" and "y" are further split into training and test sets:

- X_train and y_train: learning vectors;
- X_test and y_test: model validation vectors.

The process carried out is iterative, and for each prediction, the vectors are divided to use 80% of the total data for training and the remaining 20% for testing. We are in presence of a regression problem, and the forecasting process involves the following cyclical steps whose number corresponds to the number of elements in the test set:

- Model training using training data (X_train and y_train);

- Predicting the first element in the test data (y_test[0]);
- Evaluating the error as the deviation between the predicted value and the real value;
- Augmenting the training vectors with the real data sample of the sample just predicted (the first value of X_test and y_test are, respectively, added to X_train and y_train).

Therefore, for each forecast, the training vectors are augmented with the real data of the previous forecast. This means that after each prediction, the training set grows by one element (represented by the real value of the predicted element). Furthermore, the process is sustainable. As soon as a new sample is inserted into the data structure, a new prediction will be automatically forecasted.

The process is illustrated in Figure 3. At step 0, we need to forecast the first element in the test. After we perform the forecast (step 1), we augment the training set with the real value of the predicted element. This concept fits perfectly with that of real-time data processing.

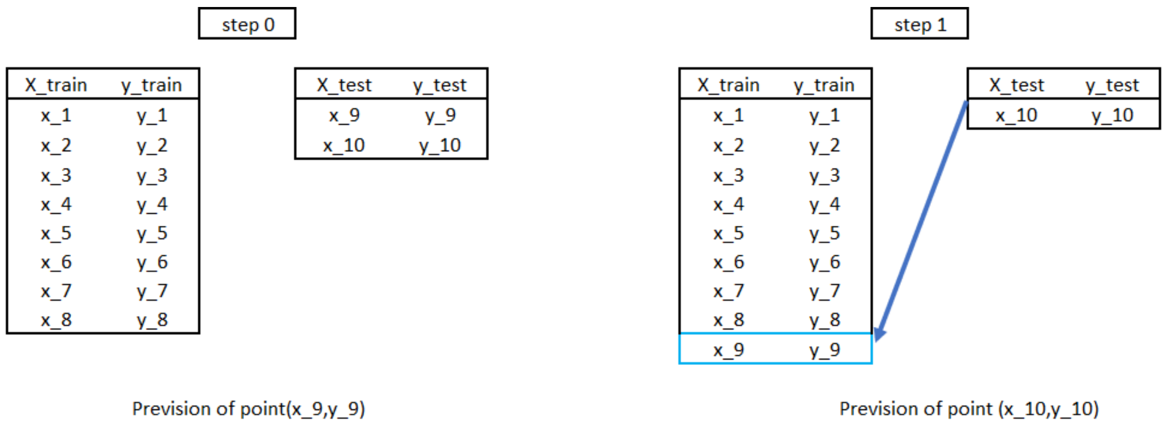

Figure 3. Example of forecast.

The regressors that will be used in this work are the following:

- Linear regression;
- ARIMA;
- Regressor based on polynomial integration.

Regression models are very popular because they adapt quickly to the data and are easy to interpret.

5.4.1. Linear Regression

Simple linear regression is nothing more than the approximation of the data along a line such as:

$$y = ax + b$$

where the coefficient "a" represents the slope, while the term "b" represents the intercept value [67].

The goal of a simple linear regression model is to replicate the existing relationships between a single characteristic defined by the independent variable x and a continuous evaluation response, also called the target variable y [68].

To use the linear regression estimator, we adopted the linear regression model from the Scikit Learn library [67].

LinearRegression fits a linear model to minimize the residual sum of squares between the observed targets in the dataset and the predicted targets from linear approximation [69].

5.4.2. ARIMA

The ARIMA model (or integrated autoregressive models with moving average) by Box and Jenkins [70], on the other hand, starts from the assumption that between two observations of a series, the one that alters the level of the series is called the disturbance. A general Box–Jenkins model is indicated as ARIMA (p, d, q) where AR = autoregression (autoregression), p is the order of AR, I = integration (integration), d is the order of I, MA = moving average and q is the order of MA. If the series is not stationary, that is, if the mean and the variance are not constant over time, it is integrated after having performed a possible transformation of the data, usually of the logarithmic type. In this way, a stationary series is obtained. The proposed procedure of Box and Jenkins is iterative and is used for the identification, estimation, and verification of an ARIMA model. It has the purpose of building a model that adapts to the observed time series and represents the generating process of the series itself:

1. Verification of the stationarity of the series;
2. Identification of the ARIMA model;
3. Estimation of the parameters p, d and q;
4. Verification of the model.

If the estimated model passes the verification stage, then it can be used for predictions. Otherwise, the identification, estimation, and verification phases are repeated iteratively [70].

In our application case, this procedure was automated using auto-ARIMA, which provides an automatic ARIMA algorithm. The auto-ARIMA process tries to identify the most optimal parameters for an ARIMA model. The problem with this automatic model is that due to stationarity problems, it may not find a suitable ARIMA model that will converge. The reason why we chose an automatic model lies in the desire to find a simple procedure that can be used without technical expertise on ARIMA. In fact, with this process, there is the risk of making the series excessively stationary, affecting the final results. Therefore, it was decided to initialize auto-ARIMA with the standard parameters proposed by the library itself [71].

5.4.3. Regressor Based on Polynomial Integration

The last model is an algorithm that we designed and created from scratch based on polynomial interpolation. The idea is to look for the polynomial of degree p that best approximates the time series in question. To do this, every time a prediction is performed, several polynomials with degree p ranging from 1 to n are generated, where n is the total number of polynomials that are thought to fit. In our case, we have chosen not to go beyond the ninth degree for each prediction; therefore, we chose among 9 polynomials. This choice was made because polynomials with a very high degree could be problematic to converge [72]. To choose the best polynomial, we carried out the following procedure:

- We determined the prediction in the point preceding the one we are interested in making the prediction for;
- The error was calculated as the difference between the obtained element and its corresponding real value;
- We saved the error in a list.

In this process, it was necessary to impose two constraints:

- The first constraint requires that the forecast value (which is a number corresponding to a date) must always be greater than the date corresponding to the previous element, as it cannot be possible to obtain forecasts in the past;
- The second constraint imposes that the failure date must not tend towards an infinite value.

This process is applied to all the polynomials and the one that at step i-1 provides the lowest error (and satisfies the two constraints defined above) is chosen to predict step i-th. This process tries to optimize the forecast by trying to follow the trend as much as possible.

An example of how it works is included in Figure 4, where we have four points in the training set and are interested in the prediction of the fifth one (in purple). Three polynomials with a different degree are determined on the first three points. The predictions on the fourth point (the three blue points are the predictions, whereas the red point is the real value) are performed and the three errors are computed. The polynomial corresponding to the lowest error, *e*3, is yellow, and it will be used to predict the fifth point.

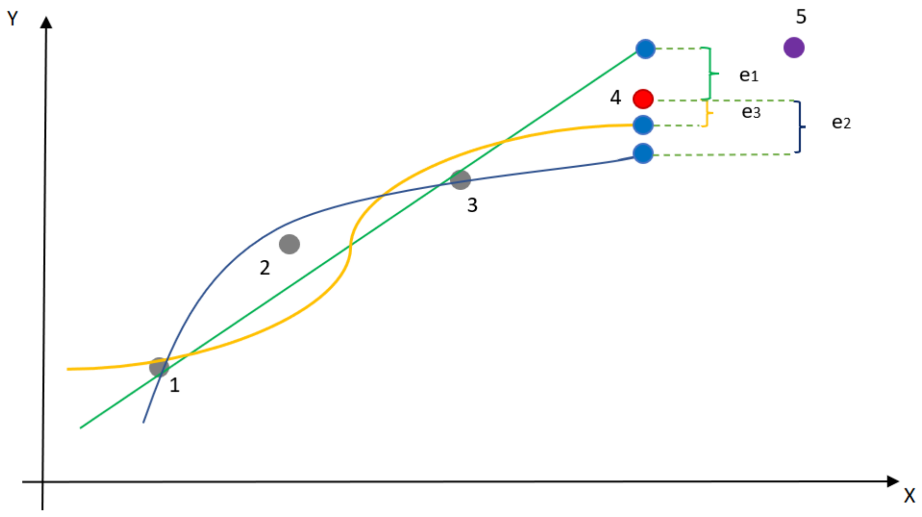

Figure 4. Example of how the polynomial integration works.

6. Experimental Evaluation

The results obtained during the whole process described in Section 5 will be shown in this section. First of all, in Section 6.1 the results related to the reconstruction of unstructured data are highlighted, showing the information related to each reconstruction strategy. Subsequently, in Section 6.2, we will report the results inherent to the techniques for the categorization of the alerts. Finally, in Section 6.3, we will discuss the results related to the forecasting of faults obtained by means of different forecasting approaches.

6.1. Results on Data Reconstruction

The "Notice" data contains 10,700 alarms related to malfunction and/or breakdown. By checking the technical site of each notice, it emerged that 10,567 assets had a missing or incomplete technical site. Consequently, it was necessary to define algorithms or completion rules. As already mentioned above, there are five of these rules, and for each of them, we report in Table 11 the number of technical sites that have been augmented. The reader notices that the same technical site may have been augmented by more than one strategy.

The number of technical sites that have been augmented is 7204 over a total of 10,700. This corresponds to having reconstructed 68.2% of the input samples.

The reconstruction results were manually validated in collaboration with the employees of ITALTELECO, who labeled as correct 100% of the performed augmentations. The reader notices that it has not been possible to automatically reconstruct a higher number of technical locations due to a lack of information necessary for their reconstruction. The lacking of a structured process when filling the notice reports led to several missing data.

6.2. Categorization of the Alerts

The entire process of categorization has already been discussed in Section 5.2, where it is reported that the frequency of words has been computed according to each type of component and overall.

The reason to distinguish between these two cases is that certain words should occur more often in certain types of components. For example, we expect that the word "LEAK" is important in those types of components that may have fluid leaks, such as pumps, and it will be less important in those components that are not subject to this type of phenomenon.

Table 11. Results of unstructured data reconstruction.

Strategy	Name	Number of Reconstructed Technical Sites
1	Completion using the information contained in the "Equipment" column of "Notices"	2598
2	Completion using the information contained in the "Equipment" column of "ODM"	104
3	Completion through the technical sites of the "ODM" data	2602
4	Completion by mapping the words contained in the "Description" column and the names of the components	1650
5	Search for components using the suffixes of the components contained in the "Description" field	196

By carrying out the categorization for each alert, we were able to categorize only 3641 alerts, which corresponds to 0.34% of the overall number of them. This low percentage strongly depends on how the descriptions of the warnings are written. As already mentioned, they suffer from the problem of having too little information. As an example, the description "RT2 MP5B NO WATER CIRCULATION TXT", in the second-to-last row of Table 9 and related to a pump, has the expression "NO WATER CIRCULATION", which is too general to be assigned to one of the categories of Table 8. It may be associated with faults related to DAMAGE, FUNCTIONALITY, REGOLATION, LEAK, REPAIR, or REPLACEMENT.

In Figure 5, the percentages of occurrences of the fault categories for all the components are indicated. Figures 6 and 7 show the frequency distribution of the fault categories of compressors and pumps, respectively.

Clearly, the two graphs show a different distribution of the fault categories. This is expected, as different component types may be subject to different failures. More specifically, we notice that the pumps are more subject to fluid loss phenomena (LEAK), perhaps due to the rupture of pipes. Conversely, the compressors are more subject to failures related to cleaning (CLEANING), perhaps linked to the clogging of pipes or filters.

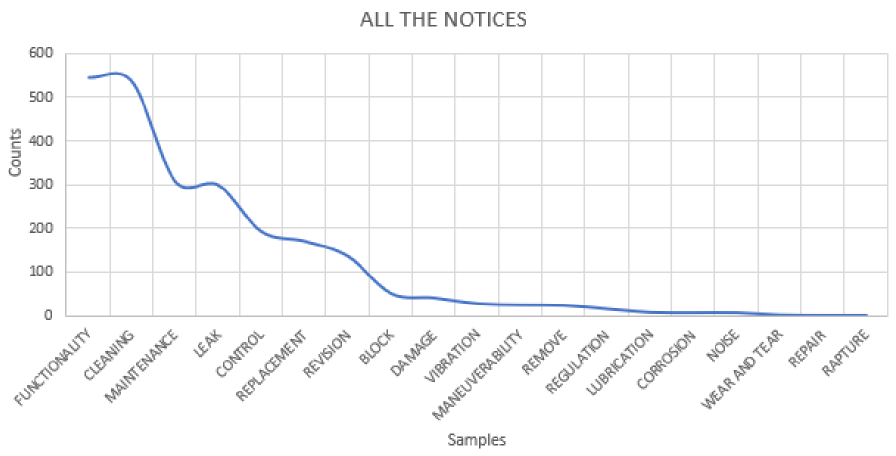

Figure 5. Frequency distribution of the fault category with respect to the notices of all the components.

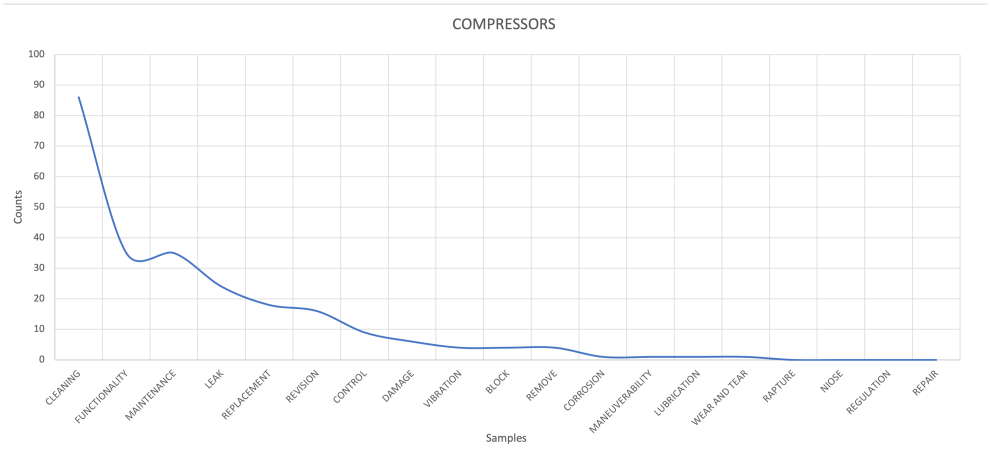

Figure 6. Frequency distribution of the fault category with respect to the notices related to compressors.

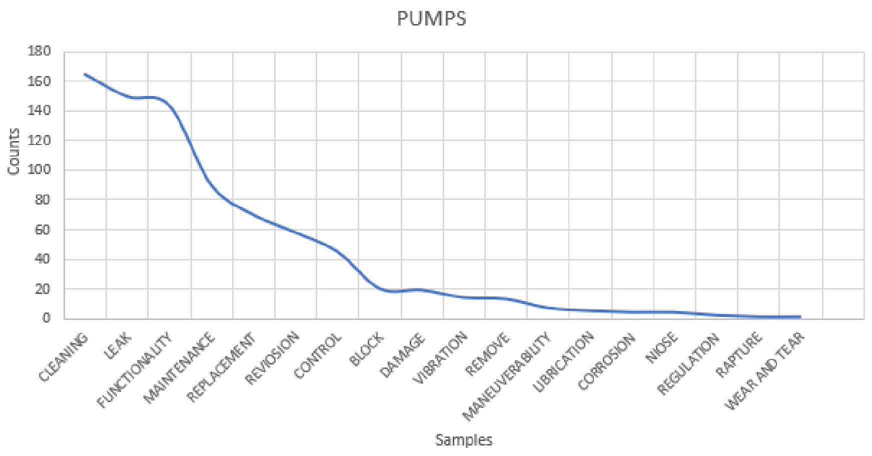

Figure 7. Frequency distribution of the fault category with respect to the notices related to pumps.

6.3. Results on Alerts Forecasting

Once the notices were reconstructed, we grouped them by component. Each group was reordered in ascending chronological order. In this way, each group corresponded to a data structure that we generated that was used for the predictive models. For each component we have analysed, we split our process into two parts:

- Extraction of the time series of faults and creation of training data and test data;
- Predictive phase.

We have already discussed the first part in Section 5.4, where we have also mentioned the three models that we have used for the forecasting phase.

We are targeting a regression problem, and the prediction we will obtain is represented by a number corresponding to an elapsed interval time. This is considered good and reliable from ITALTELECO if it is less or equal than three months from the real value. Therefore, in the rest of this section, this will be our reference threshold.

To let the reader understand the process of our evaluation, let us consider the RT2MP4-P component, which corresponds to a pump. For this component, there are 43 warnings relating to faults. These were transformed into a vector of dates ordered chronologically. This vector was then divided into two parts.

- A total of 80% of the data (corresponding to 34 faults for the example above) became the training vector;
- The remaining 20% (corresponding to 8 faults) were used for the test phase.

The training vector was fed to the three predictive models introduced in Section 5.4. Figure 8 shows the forecast of the eight elements in the test set performed by the linear regression. The gray points represent the training data, the red point is the forecasted element, and the blue element is its corresponding real value. The reader notices that the red point at step i becomes a new training element in the next iteration (the last gray point in the graph at step $i + 1$). Moreover, the slope of the forecast line, shown in green, changes as we add new elements in the training data as the course of the line tends to follow the real data. The errors are computed as the absolute difference on the y-axis between the forecasted point (red) and the real one (blue). Related to Figure 8, Table 12 illustrates the value of the y coordinates of the eight points of the test set, the predicted value using the linear regression model, and the error.

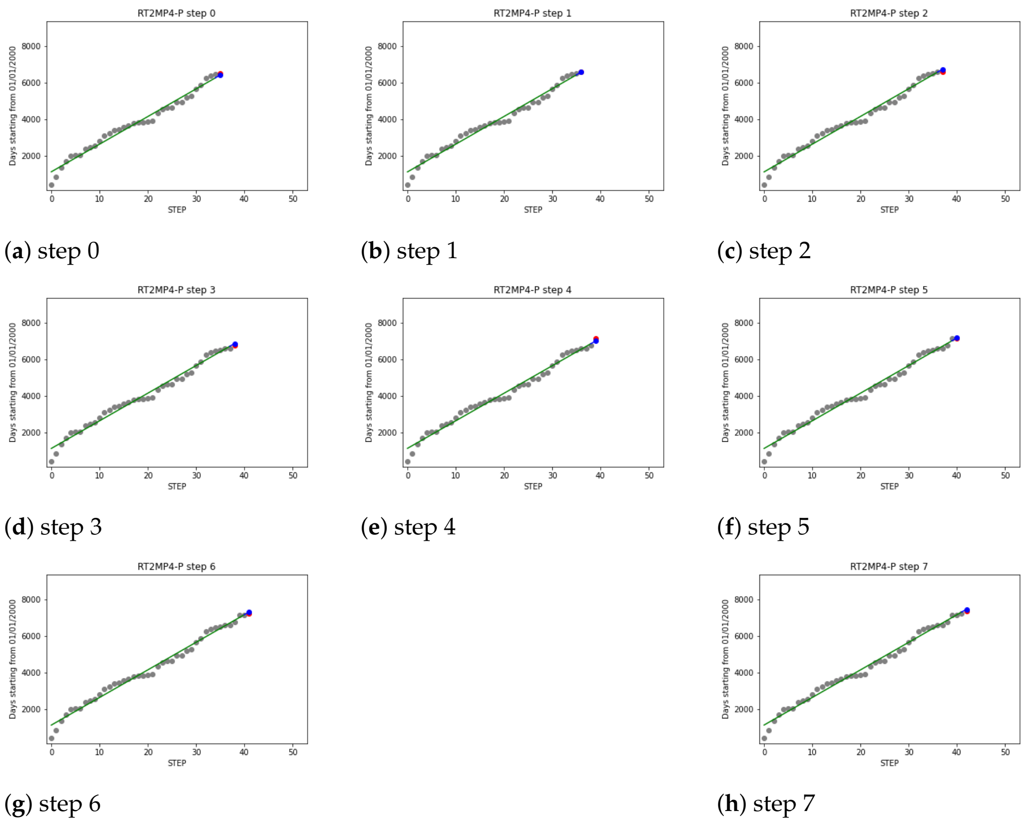

Figure 8. Step by step predictions performed by the linear regression model of the RT2MP4-P component.

Table 12. Results table of the RT2MP4-P component performed by the linear regression Model.

#Step	y_TEST	Predict Value LR	Error LR
step 0	6510	6413.62521	96.37
step 1	6578	6575.287302	2.71
step 2	6611	6726.968468	115.96
step 3	6744	6866.16074	122.16
step 4	7129	7004.561404	124.43
step 5	7136	7167.465385	31.46
step 6	7249	7315.310976	66.31
step 7	7375	7459.801394	84.80
Average error			*95.02*

We applied the three regression models to the alerts of 181 system components, and the results are shown in Table 13.

Table 13. Average error of each regression model.

Average Error LR	Average Error ARIMA	Average Error PI
647.66	418.56	476.15

The overall average error is very high: more than 647 days for the linear regression, more than 418 days for ARIMA and more than 476 days for the polynomial integration method. This phenomenon is due to the large amount of components having a very small number of notices. In fact, we analyzed 181 out of the overall number of 1,036 components because of the insufficient number of samples found in the "Notice" data to be applied to the models. In particular, we processed components with a number of notices higher than 3 for the linear regression and the polynomial integration method. ARIMA needed at least 10 samples.

ITALTELECO argues that an acceptable error should fall within three months. The reader can observe that the obtained error is much higher than the desired one (3 months corresponding to 90 days). As several components had a few number of samples to be used as training set, the predictions in such cases are usually very far from the real values. Therefore, we counted the number of samples for each component, and Table 14 indicates the related distribution and the corresponding errors averaged per each model and the components with a number of notices fixed in the first column. The reader notices that the average error for ARIMA for the first six rows is empty because the model requires at least 10 samples, as previously mentioned.

Therefore, we analysed the performances with respect to the sample size and identified the cases that could generate an acceptable error. We defined the global error E_c^M as the errors of the model M computed, taking all the errors corresponding to the components whose number of samples was greater than c. For example, E_{10}^{LR} corresponds to the average error of the linear regression for all the components with number of samples greater than 10 (from sample length 11 on). Table 15 illustrates the global error E_c for different values of c. As indicated in Table 14, by choosing c = 43, we obtained the minimum average error of 84.11 days. From this, it can be deduced that the model that outperforms the others is ARIMA, and that for a sample size greater than 43 elements, both ARIMA and PI satisfy the constraint related to the maximum acceptable error.

Table 14. Distributions of errors per model and per components with the same number of samples.

Sample Length	#Components with #Samples Defined in the First Column	Average Error LR	Average Error ARIMA	Average Error PI
4	52	1104.97		1062.90
5	17	819.59		1149.15
6	14	793.61		813.02
7	12	949.85		1052.04
8	7	790.84		799.89
9	6	548.83		377.57
10	8	1037.15	1104.17	1198.78
11	9	802.77	823.65	634.69
12	5	432.93	458.22	458.83
13	3	412.57	418.71	685.24
14	1	3108.11	1787.67	1214.57
15	2	284.70	419.67	423.10

Table 14. *Cont.*

Sample Length	#Components with #Samples Defined in the First Column	Average Error LR	Average Error ARIMA	Average Error PI
16	3	747.45	788.68	616.79
17	8	784.69	501.16	504.10
18	5	1141.74	762.67	745.30
19	3	942.11	1075.21	545.53
20	1	285.41	383.63	453.06
21	4	829.39	484.20	548.16
23	1	276.26	273.32	303.25
25	2	1011.60	462.71	324.45
26	1	367.04	257.56	258.31
27	1	205.67	216.57	257.99
28	2	255.18	120.25	193.06
32	1	412.46	170.53	253.03
37	1	429.07	328.17	391.99
40	1	157.14	65.31	113.24
43	3	427.17	229.02	152.33
47	1	152.12	159.03	118.26
52	2	282.07	103.39	117.10
53	1	124.62	109.53	135.59
110	1	655.20	73.42	81.05
148	1	297.10	45.29	46.17
161	1	105.62	42.38	53.58
198	1	1045.45	55.72	106.89

Table 15. Global errors E_c^M for different values of c.

Number of Samples	Average Error LR	Average Error ARIMA	Average Error PI
10	591.69	393.17	360.58
21	387.74	169.51	181.64
43	386.17	84.11	94.09

7. Conclusions and Future Works

This paper provides a methodology to preprocess time series data related to fault components of one of the systems of the SARAS company (oil and gas sector) aiming at implementing a machine learning model to predict the next failure date of each component of such a system. We performed a data cleaning step to fill in several of the missing information within the raw data. This phase has been performed together with ITALTELECO, a company partner of SARAS, which provided us with the necessary information to rebuild some of the technical sites needed for the prediction step. Then, we identified the category of the monitored faults for each component through natural language processing and clustering techniques. An accurate analysis of the registered failures led to the identification of 19 fault categories, and, for each of them, an unique identifier i.e., the notice, was assigned. Then, we tried to predict the next fault occurrences for each component. This has been mapped as a regression problem where the number to predict corresponded to a date (number of days elapsed from 31/12/1999). Three different machine learning models have been used to tackle the regression problem. Time series data of notices were used to train each model, and the performances of each model have been analysed.

To support machine learning models in making reliable predictions, it is important to make sure there are enough complete data. One of the most problematic parts of our work was related to the reconstruction of missing information. The following issues were found among the data and have been highlighted:

- incorrect compilation of a description where several different components were often inserted;
- an incorrect compilation of the component names, i.e., the component names in the description are often named without the RT2 prefix or the last word indicating the real part of the component subject to a maintenance event;
- lack of important data such as downtime that would have been useful to try to predict data related to lost production;
- the component map provided was not complete and therefore it was not possible to recognize some components and reconstruct their technical locations.

To solve some of these problems, natural language processing techniques were used to recognize components using only parts of their name. In this way, if a word in the description contained a mapped component, it has been recognized, and, consequently, it has been possible to reconstruct the technical location of the relevant warning. Regarding the data on downtime and maintenance costs, nothing could be done to reconstruct them. The only action was to provide feedback to the company showing them the importance of including this type of information in future actions.

One of the best practices that come from our analysis is to start collecting data using a schema defined at the beginning of the collection process or to set up a system capable of acquiring data directly without having to post-process them by hand. This would bring the following advantages:

- No pre-processing of the initial data;
- Better data quality;
- Better performance of the predictive models.

It would be advisable to use special hardware systems equipped with ad hoc software that guides the workers during maintenance activities to correctly fill the report. Drop-down menus or other easy-to-use techniques that can facilitate the acquisition phase of the data should be adopted. For example, these menus should contain standard words that the operator can easily choose to speed up the writing process and reduce errors when entering information. In this way, the history of these activities will be clean and complete.

This would improve the forecast performances, and the better the forecast is, the lower the costs will be, as the used components will be exploited throughout their life. This will further avoid the waste of unnecessary resources, the costs related to the replacement or repair of some components, and the loss of production due to the unexpected shutdown of the plant. At this time, the forecasts obtained are subject to particularly high average errors, and only in cases where samples were large enough, we obtained acceptable predictions.

On the other hand, potential limitations of our approach are related to the peculiarity of the used data: the methodology is applicable to other domains, but each use case will be different, and a human component must be always employed to identify the features of the data that need to be completed.

As future directions, we would like to improve the prediction procedure by integrating sensor data. This is important because the proposed procedure does not take into account the working conditions of the individual component. The sensors provide useful quantities to understand the operating status of each component. Vibrations are, in fact, one of the physical quantities that can be analyzed for this purpose by indicating the state of health of a plant component by signaling anomalies when the values go out of the predetermined ranges.

Moreover, the generated data structure could also be automated with an IoT system and data storage in the Cloud with automatic procedures. In this way, it will be possible to train the machine learning models and make predictions in real time, and this will allow us to intervene promptly in the event of an imminent failure. Thanks to the help of these models, it will therefore be possible to intervene when strictly necessary to optimize the number of maintenance interventions and exploit the underlying component as much as possible.

Author Contributions: Conceptualization, P.F.O., G.M. and S.M.; methodology, P.F.O. and D.R.R.; software, R.P. and G.M.; validation, R.P. and D.R.R.; formal analysis, D.R.R. and S.A.; Investigation, P.F.O. and S.A.; Resources, G.M. and S.M.; data curation, R.P., G.M. and S.M.; writing original draft preparation, R.P. and D.R.R.; writing review and editing, D.R.R., S.A.; supervision, P.F.O. and D.R.R.; project administration, P.F.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Acronym	**Description**
a	Slope coefficient
ANFIS	Adaptive neuro-fuzzy inference systems
ANN	Artificial neural networks
AR	Autoregression
ARIMA	Auto-regressive integrated moving average
b	Intercept value
c	Sample length
CNN	Convolutional neural network
d	Order of integration
DGA	Dissolved gas analysis
e	Error
E	Global error
ELM	Extreme learning machine
HSA	Hierarchical symbolic analysis
HVAC	Heating ventilation and air conditioning
i	Number of steps
I	Integration
LSTM	Long-short-term memory
LPG	Liquefied petroleum gas
LR	Linear regression
M	Model
MA	Moving average
ML	Machine learning
MLP	Multi-layered perceptron
n	Total number of polynomials
NLP	Simple natural language processing
ODM	Maintenance order
p	Order of autoregression
PdM	Predictive maintenance
PI	Polynomial integration
PvM	Preventive maintenance
q	Order of moving average
R2F	Run to failure

RT2	Revamping topping plant
SVM	Support-vector machines
x	Independent variable
y	Target variable

References

1. Orrù, P.F.; Zoccheddu, A.; Sassu, L.; Mattia, C.; Cozza, R.; Arena, S. Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. *Sustainability* **2020**, *12*, 4776. [CrossRef]
2. Sajid, S.; Haleem, A.; Bahl, S.; Javaid, M.; Goyal, T.; Mittal, M. Data science applications for predictive maintenance and materials science in context to Industry 4.0. *Mater. Today Proc.* **2021**, *45*, 4898–4905. [CrossRef]
3. Bertolini, M.; Mezzogori, D.; Neroni, M.; Zammori, F. Machine Learning for industrial applications: A comprehensive literature review. *Expert Syst. Appl.* **2021**, *175*, 114820. [CrossRef]
4. Consoli, S.; Recupero, D.R.; Saisana, M. *Data Science for Economics and Finance: Methodologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–355. [CrossRef]
5. Consoli, S.; Recupero, D.R.; Petković, M. *Data Science for Healthcare: Methodologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–367. [CrossRef]
6. Susto, G.A.; Schirru, A.; Pampuri, S.; Pagano, D.; McLoone, S.F.; Beghi, A. A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation. In Proceedings of the Ninth International Conference on Automation Science and Engineering, Madison, WI, USA, 17–20 August 2013; pp. 195–200. [CrossRef]
7. Mönch, L.; Fowler, J.; Dauzère-Pérès, S.; Mason, S.; Rose, O. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *J. Sched.* **2011**, *14*, 583–599. [CrossRef]
8. Köksal, G.; İnci Batmaz.; Testik, M.C. A review of data mining applications for quality improvement in manufacturing industry. *Expert Syst. Appl.* **2011**, *38*, 13448–13467. [CrossRef]
9. Wang, Y.; Xue, C.; Jia, X.; Peng, X. Fault diagnosis of reciprocating compressor valve with the method integrating acoustic emission signal and simulated valve motion. *Mech. Syst. Signal Process.* **2015**, *56*, 197–212. [CrossRef]
10. Susto, G.A.; Schirru, A.; Pampuri, S.; McLoone, S.; Beghi, A. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Trans. Ind. Inform.* **2015**, *11*, 812–820. [CrossRef]
11. Leukel, J.; González, J.; Riekert, M. Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *J. Manuf. Syst.* **2021**, *61*, 87–96. [CrossRef]
12. Sahal, R.; Breslin, J.G.; Ali, M.I. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. *J. Manuf. Syst.* **2020**, *54*, 138–151. [CrossRef]
13. Wang, J.; Xu, C.; Zhang, J.; Zhong, R. Big data analytics for intelligent manufacturing systems: A review. *J. Manuf. Syst.* **2022**, *62*, 738–752. [CrossRef]
14. Rožanec, J.M.; Zajec, P.; Kenda, K.; Novalija, I.; Fortuna, B.; Mladenić, D.; Veliou, E.; Papamartzivanos, D.; Giannetsos, T.; Menesidou, S.A.; et al. STARdom: An Architecture for Trusted and Secure Human-Centered Manufacturing Systems. In Proceedings of the Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems, Nantes, France, 5–9 September 2021; Dolgui, A., Bernard, A., Lemoine, D., von Cieminski, G., Romero, D., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 199–207.
15. Mobley, R.K. *An Introduction to Predictive Maintenance*; Elsevier: Amsterdam, The Netherlands, 2002.
16. Palacín, I.; Gibert, D.; Planes, J.; Arena, S.; Orrù, P.F.; Melis, M.; Annis, M. Anomaly Detection for Diagnosing Failures in a Centrifugal Compressor Train. In Proceedings of the Artificial Intelligence Research and Development—Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, CCIA 2021, Virtual Event, 20–22 October 2021; Villaret, M., Alsinet, T., Fernández, C., Valls, A., Eds.; Volume 339: Frontiers in Artificial Intelligence and Applications; IOS Press: Amsterdam, The Netherlands, 2021; pp. 217–220. [CrossRef]
17. Liu, R.; Yang, B.; Zio, E.; Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.* **2018**, *108*, 33–47. [CrossRef]
18. Li, K.; Xiong, M.; Li, F.; Su, L.; Wu, J. A novel fault diagnosis algorithm for rotating machinery based on a sparsity and neighborhood preserving deep extreme learning machine. *Neurocomputing* **2019**, *350*, 261–270. [CrossRef]
19. Zhang, Z.; Li, S.; Wang, J.; Xin, Y.; An, Z. General normalized sparse filtering: A novel unsupervised learning method for rotating machinery fault diagnosis. *Mech. Syst. Signal Process.* **2019**, *124*, 596–612. [CrossRef]
20. Luciano Furlanetto, M.G.; Macchi, M. *Principi Generali di Gestione Della Manutenzione*; FrancoAngeli: Milan, Italy, 2006.
21. Babai, M.; Ali, M.; Boylan, J.; Syntetos, A. Forecasting and inventory performance in a two-stage supply chain with ARIMA(0,1,1) demand: Theory and empirical analysis. *Int. J. Prod. Econ.* **2013**, *143*, 463–471. [CrossRef]
22. Duan, Y.; Wang, H.; Wei, M.; Tan, L.; Yue, T. Application of ARIMA-RTS optimal smoothing algorithm in gas well production prediction. *Petroleum* **2022**, *8*, 270–277. [CrossRef]
23. Zhu, J.; Ge, Z.; Song, Z.; Gao, F. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annu. Rev. Control* **2018**, *46*, 107–133. [CrossRef]
24. Ning, Y.; Kazemi, H.; Tahmasebi, P. A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet. *Comput. Geosci.* **2022**, *164*, 105126. [CrossRef]

25. Singh, V.; Mathur, J.; Bhatia, A. A Comprehensive Review: Fault Detection, Diagnostics, Prognostics, and Fault Modelling in HVAC Systems. *Int. J. Refrig.* 2022, *in press*.
26. Dogan, A.; Birant, D. Machine learning and data mining in manufacturing. *Expert Syst. Appl.* **2021**, *166*, 114060. [CrossRef]
27. Seo, B.; Shin, J.; Kim, T.; Youn, B.D. Missing data imputation using an iterative denoising autoencoder (IDAE) for dissolved gas analysis. *Electr. Power Syst. Res.* **2022**, *212*, 108642. [CrossRef]
28. Wang, Y.; Wang, J.; Li, Z.; Yang, H.; Li, H. Design of a combined system based on two-stage data preprocessing and multi-objective optimization for wind speed prediction. *Energy* **2021**, *231*, 121125. [CrossRef]
29. Ranjan, K.G.; Prusty, B.R.; Jena, D. Review of preprocessing methods for univariate volatile time-series in power system applications. *Electr. Power Syst. Res.* **2021**, *191*, 106885. [CrossRef]
30. He, R.; Xiao, T.; Qiu, S.; Gu, J.; Wei, M.; Xu, P. A rule-based data preprocessing framework for chiller rooms inspired by the analysis of engineering big data. *Energy Build.* **2022**, *273*, 112372. [CrossRef]
31. Xiao, X.; Xiao, Y.; Zhang, Y.; Qiu, J.; Zhang, J.; Yildirim, T. A fusion data preprocessing method and its application in complex industrial power consumption prediction. *Mechatronics* **2021**, *77*, 102520. [CrossRef]
32. Xiao, Z.; Gang, W.; Yuan, J.; Chen, Z.; Li, J.; Wang, X.; Feng, X. Impacts of data preprocessing and selection on energy consumption prediction model of HVAC systems based on deep learning. *Energy Build.* **2022**, *258*, 111832. [CrossRef]
33. Frye, M.; Mohren, J.; Schmitt, R.H. Benchmarking of Data Preprocessing Methods for Machine Learning-Applications in Production. *Procedia CIRP* **2021**, *104*, 50–55. [CrossRef]
34. Status of Data-Driven Methods and their Applications in Oil and Gas Industry, Volume Day 3 Wed, SPE Europec featured at EAGE Conference and Exhibition. 2018. p. D031S005R007. Available online: http://xxx.lanl.gov/abs/https://onepetro.org/SPEEURO/proceedings-pdf/18EURO/3-18EURO/D031S005R007/1209097/spe-190812-ms.pdf (accessed on 11 June 2018). doi: 10.2118/190812-MS. [CrossRef]
35. Eken, S. An exploratory teaching program in big data analysis for undergraduate students. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 4285–4304. [CrossRef]
36. Dalzochio, J.; Kunst, R.; Pignaton, E.; Binotto, A.; Sanyal, S.; Favilla, J.; Barbosa, J. Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. *Comput. Ind.* **2020**, *123*, 103298. [CrossRef]
37. Carvalho, T.P.; Soares, F.A.; Vita, R.; Francisco, R.d.P.; Basto, J.P.; Alcalá, S.G. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106024. [CrossRef]
38. Hajizadeh, Y. Machine learning in oil and gas; a SWOT analysis approach. *J. Pet. Sci. Eng.* **2019**, *176*, 661–663. [CrossRef]
39. Hanga, K.M.; Kovalchuk, Y. Machine learning and multi-agent systems in oil and gas industry applications: A survey. *Comput. Sci. Rev.* **2019**, *34*, 100191. [CrossRef]
40. Qian, W.; Li, S.; Yi, P.; Zhang, K. A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions. *Measurement* **2019**, *138*, 514–525. [CrossRef]
41. Zhang, W.; Li, X.; Ding, Q. Deep residual learning-based fault diagnosis method for rotating machinery. *ISA Trans.* **2019**, *95*, 295–305. [CrossRef]
42. Yang, Y.; Zheng, H.; Li, Y.; Xu, M.; Chen, Y. A fault diagnosis scheme for rotating machinery using hierarchical symbolic analysis and convolutional neural network. *ISA Trans.* **2019**, *91*, 235–252. [CrossRef]
43. Qian, W.; Li, S.; Wang, J.; Wu, Q. A novel supervised sparse feature extraction method and its application on rotating machine fault diagnosis. *Neurocomputing* **2018**, *320*, 129–140. [CrossRef]
44. Yang, B.; Lei, Y.; Jia, F.; Xing, S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mech. Syst. Signal Process.* **2019**, *122*, 692–706. [CrossRef]
45. Erdem, T.; Eken, S. Layer-Wise Relevance Propagation for Smart-Grid Stability Prediction. In Proceedings of the Pattern Recognition and Artificial Intelligence, Xiamen, China, 23–25 September 2022; Djeddi, C., Siddiqi, I., Jamil, A., Ali Hameed, A., Kucuk, İ., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 315–328.
46. Breviglieri, P.; Erdem, T.; Eken, S. Predicting Smart Grid Stability with Optimized Deep Models. *SN Comput. Sci.* **2021**, *2*, 73. [CrossRef]
47. Bilski, P. Application of support vector machines to the induction motor parameters identification. *Measurement* **2014**, *51*, 377–386. [CrossRef]
48. Zhang, X.; Chen, W.; Wang, B.; Chen, X. Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization. *Neurocomputing* **2015**, *167*, 260–279. [CrossRef]
49. Panda, A.K.; Rapur, J.S.; Tiwari, R. Prediction of flow blockages and impending cavitation in centrifugal pumps using Support Vector Machine (SVM) algorithms based on vibration measurements. *Measurement* **2018**, *130*, 44–56. [CrossRef]
50. Guedes, A.S.; Silva, S.M.; de Jesus Cardoso Filho, B.; Conceicao, C.A. Evaluation of electrical insulation in three-phase induction motors and classification of failures using neural networks. *Electr. Power Syst. Res.* **2016**, *140*, 263–273. [CrossRef]
51. Yu, S.; Zhao, D.; Chen, W.; Hou, H. Oil-immersed power transformer internal fault diagnosis research based on probabilistic neural network. *Procedia Comput. Sci.* **2016**, *83*, 1327–1331. [CrossRef]
52. Romeo, L.; Loncarski, J.; Paolanti, M.; Bocchini, G.; Mancini, A.; Frontoni, E. Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. *Expert Syst. Appl.* **2020**, *140*, 112869. [CrossRef]

53. Giantomassi, A.; Ferracuti, F.; Iarlori, S.; Ippoliti, G.; Longhi, S. Electric motor fault detection and diagnosis by kernel density estimation and Kullback–Leibler divergence based on stator current measurements. *IEEE Trans. Ind. Electron.* **2014**, *62*, 1770–1780. [CrossRef]
54. Chen, Z.; Gryllias, K.; Li, W. Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mech. Syst. Signal Process.* **2019**, *133*, 106272. [CrossRef]
55. Wang, X.B.; Zhang, X.; Li, Z.; Wu, J. Ensemble extreme learning machines for compound-fault diagnosis of rotating machinery. *Knowl.-Based Syst.* **2020**, *188*, 105012. [CrossRef]
56. Pang, S.; Yang, X.; Zhang, X.; Lin, X. Fault diagnosis of rotating machinery with ensemble kernel extreme learning machine based on fused multi-domain features. *ISA Trans.* **2020**, *98*, 320–337. [CrossRef]
57. Wang, Z.Y.; Lu, C.; Zhou, B. Fault diagnosis for rotary machinery with selective ensemble neural networks. *Mech. Syst. Signal Process.* **2018**, *113*, 112–130. [CrossRef]
58. Khorsheed, R.M.; Beyca, O.F. An integrated machine learning: Utility theory framework for real-time predictive maintenance in pumping systems. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2021**, *235*, 887–901. [CrossRef]
59. Tang, S.; Shen, C.; Wang, D.; Li, S.; Huang, W.; Zhu, Z. Adaptive deep feature learning network with Nesterov momentum and its application to rotating machinery fault diagnosis. *Neurocomputing* **2018**, *305*, 1–14. [CrossRef]
60. Zenisek, J.; Holzinger, F.; Affenzeller, M. Machine learning based concept drift detection for predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106031. [CrossRef]
61. Lee, W.J.; Wu, H.; Yun, H.; Kim, H.; Jun, M.B.; Sutherland, J.W. Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. *Procedia Cirp* **2019**, *80*, 506–511. [CrossRef]
62. Mohammadpoor, M.; Torabi, F. Big Data analytics in oil and gas industry: An emerging trend. *Petroleum* **2020**, *6*, 321–328. [CrossRef]
63. Ruiz-Sarmiento, J.R.; Monroy, J.; Moreno, F.A.; Galindo, C.; Bonelo, J.M.; Gonzalez-Jimenez, J. A predictive model for the maintenance of industrial machinery in the context of industry 4.0. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103289. [CrossRef]
64. Servizio del Personale. Manuale Introduttivo alla Raffineria. 1–166. Internal Technical Report. *Documento Saras*.
65. Servizio dei Processi. Uno Sguardo sul Mondo della Raffinazione del Petrolio, 1–101. Internal Technical Report. *Documento Saras*.
66. Saras. Rapporto di Sicurezza di Stabilimento. Volume II. Impianti di Distillazione Atmosferica Topping 1 (T1), Topping RT2 Impianto Merox Kerosene / Minalk. Aggiornamento October 2005. 1–131. Internal Technical Report. *Documento Saras*.
67. VanderPlas, L. *Python Data Science Handbook: Essential Tools for Working with Data*; O'Reilly Media: Sebastopol, CA, USA, 2016.
68. Raschka, S. *Machine Learning con Python: Costruire Algoritmi per Generare*; Apogeo Education: Milan, Italy, 2017.
69. Scikit Learn Developers. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (accessed on 1 July 2021).
70. Prabhakaran, S. Available online: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/ (accessed on 1 July 2021).
71. Sphinx. Available online: https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html (accessed on 1 July 2021).
72. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical Recipes: The Art of Scientistic Compunting*; Cambridge University Press: Cambridge, UK, 2007.